

Pembuatan Simulasi 3D Virtual Reality Berbasis Android Sebagai Alat Bantu Terapi Acrophobia

Ghali Adyo Putra^{*)}, Rinta Kridalukmana, Kurniawan Teguh Martono

Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

ABSTRACT – *Virtual Reality is three-dimensional technology and developing rapidly at the moment. Therefore, Virtual Reality technology implementation will be useful for people. One of this implementation is in clinical fields, which is for handling phobia. One of the therapies provided by a therapist to patients is by using flooding technique that faces the patient on the situation that confronts the making of fear until no longer feel anxious. In some cases, the flooding technique for acrophobia is almost impossible because of the level of dangerous and the expensive cost. By using game engine Unity and Multimedia Development Life Cycle (MDLC) method which has six stages, concept, design, material collection, assembly, testing, and distribution, application development Android-based Virtual Reality will resolve the issue. Based on the results of Black-box testing, the application of virtual reality simulation has run its functionalities as a tool for the treatment of acrophobia well. Further research is needed to find out the impact of application in the medical sector.*

Keywords – *Acrophobi; Virtual Realit; MDLC;Unity*

ABSTRAK - *Acrophobia merupakan jenis fobia yang membuat individu merasakan kegelisahan, ketegangan, dan rasa tidak nyaman ketika berada pada ketinggian. Hal itu yang membuat penderita acrophobia tidak bebas melakukan aktivitas sehari-hari. Untuk mengatasinya, dibutuhkan terapi ke psikiater untuk menyembuhkan acrophobia tersebut. Dalam teknik terapi, ada yang dinamakan flooding, yaitu menempatkan penderita ke situasi yang membuat ketakutan sampai penderita tidak merasa cemas. Teknik flooding sangat berbahaya jika diterapkan pada penderita acrophobia. Teknologi Virtual Reality sudah digunakan untuk menangani penderita acrophobia, tetapi teknologi yang ada masih mahal. Maka dari itu diperlukannya teknologi yang murah supaya penderita acrophobia dapat melakukan terapi dengan aman. Dengan menggunakan Multimedia Development Life Cycle sebagai metode pengembangannya, Unity, dan aplikasi berbasis Android dapat mengatasi masalah tersebut. Pembuatan simulasi 3D Virtual Reality sebagai alat bantu terapi acrophobia berbasis Android telah*

berhasil dibuat. Namun, diperlukan penelitian lebih lanjut untuk melihat dampak lebih lanjut pada bidang klinis.

Kata Kunci – *Acrophobia; Virtual Reality; MDLC; Unity*

I. PENDAHULUAN

Setiap orang memiliki ketakutan atau respons yang tidak menyenangkan terhadap objek-objek, situasi atau makhluk hidup. Beberapa respons ketidaknyamanan atau ketidaksukaan, bersifat umum dan merupakan hal yang tidak terlalu perlu di perhatikan. Akan tetapi, jika respons seseorang terhadap sesuatu di antara hal yang dialami tersebut terlalu jauh di luar proporsi bahaya atau sikap yang mengancam oleh stimulus, orang tersebut dikatakan memiliki fobia. Ketakutan irasional yang kuat cukup umum terjadi dalam masyarakat, hanya belum diketahui kondisi-kondisi yang menyebabkan distress atau kelemahan yang ditemukan dalam kriteria diagnostik untuk fobia spesifik. Fobia spesifik biasa dialami secara umum rata-rata dalam sampel komunitas adalah 13,2 persen. Fobia spesifik yang paling banyak diderita salah satunya adalah *acrophobia*, atau fobia ketinggian[1].

Virtual Reality adalah simulasi komputer yang dihasilkan dari lingkungan tiga dimensi, yang tampaknya sangat nyata kepada orang yang pernah mencoba teknologinya. Tujuannya adalah untuk mencapai rasa yang kuat hadir di lingkungan virtual. Pengguna teknologi Virtual Reality menggunakan alat seperti kacamata untuk melihat adegan *stereoscope* tiga dimensi. Pengguna dapat melihat sekitar dengan menggerakkan kepalanya dan berjalan-jalan dengan menggunakan kontrol tangan atau sensor gerak. Pengguna terlibat dalam suatu pengalaman yang seolah-olah ada di dunia virtual[2]. Untuk mencegah kelelahan mata, objek yang akan dilihat oleh pengguna dengan matanya dalam jangka waktu tertentu (contohnya menu, objek yang menonjol pada lingkungan) harus menghasilkan antara sekitar 0.75 sampai 3.5 meter[3].

Para peneliti dari HIT Lab sudah melakukan penelitian terhadap efek samping Virtual Reality terhadap penderita fobia. Virtual Reality terbukti efektif dalam menangani fobia laba-laba dibandingkan

*) penulis korespondensi (Ghali Adyo Putra)

Email: ghaliap@ce.undip.ac.id

pengontrolan kondisi, dibuktikan dari kuesioner pada penderita fobia, test perilaku penghindaran (seberapa dekat pasien berkeinginan untuk berinteraksi dengan tarantula sungguhan), dan rating penderita oleh klinisi dan penilai independen. Total 83% pasien yang menggunakan Virtual Reality memperlihatkan secara klinis peningkatan yang signifikan dibandingkan dengan pasien yang tidak menggunakan Virtual Reality, membuktikan Virtual Reality dapat secara efektif mengatasi fobia[4]. Penekanan pada dunia virtual terhadap penderita harus serelaks mungkin untuk meredakan kecemasan yang ada. Penelitian terhadap Virtual Reality pada *smartphone* telah dilakukan terhadap penderita gangguan kecemasan. Aplikasi yang ada digunakan sebagai alat relaksasi penderita ketika berada dirumah. Dalam penelitian tersebut berhasil mengungkapkan bahwa relaksasi dengan menggunakan Virtual Reality dapat mengurangi gangguan kecemasan[5].

Dalam penelitian ini, aplikasi yang dibuat berbeda dari penelitian sebelumnya. Aplikasi ini memberikan kelebihan-kelebihan yang ada pada dua penelitian sebelumnya, dimana aplikasi ini dibuat sebagai alat bantu terapi *acrophobia* dengan biaya yang lebih murah dan dengan kualitas yang sama dengan Virtual Reality berbasis PC. Dengan itu, penderita fobia dapat melakukan terapi secara efektif dan dengan biaya yang relatif lebih murah.

II. METODE PENGEMBANGAN SISTEM

Aplikasi ini menggunakan metodologi yang digunakan adalah Multimedia Development Life Cycle (MDLC) yang bersumber dari Luther dan sudah dimodifikasi oleh Sutopo. Metodologi pengembangan multimedia tersebut terdiri dari enam tahap, yaitu konsep (concept), desain (design), pengumpulan materi (material collecting), pembuatan (assembly), pengujian (testing), dan distribusi (distribution). Keenam tahap ini tidak harus berurutan dalam prakteknya, tahap-tahap tersebut dapat saling bertukar posisi. Meskipun begitu, tahap konsep memang harus menjadi hal yang pertama kali dikerjakan[6].

Tahap pertama adalah pembuatan konsep. Aplikasi ini memiliki konsep, yaitu memiliki beberapa skenario yang digunakan untuk simulasi berbentuk cutscene. dan pasien harus melalui skenario-skenario tersebut. Skenario dalam aplikasi ini terdiri dari skenario menaiki elevator, skenario menaiki kincir ria, skenario menaiki cable car, dan skenario menaiki tangga.

Tahap kedua adalah desain. Tahapan desain meliputi desain sistem, struktur navigasi dan perancangan storyboard. Aplikasi ini menggunakan sistem yang simpel supaya pengguna yang awam akan Virtual Reality dapat menggunakannya dengan baik. Dalam tahapan ini juga dirumuskan spesifikasi kebutuhan untuk pembuatan aplikasi seperti perangkat keras dan

perangkat lunak. Perangkat keras yang digunakan untuk membuat aplikasi ini ialah laptop Asus N56V yang memiliki RAM 4GB, monitor LCD 15 Inchi, harddisk berukuran 750GB dan kartu grafis Intel GMA 4000 + Nvidia GT630 serta smartphone dan cardboard untuk menguji aplikasi. Perangkat lunak yang digunakan ialah Unity 5.4.1, Autodesk Maya 2016, dan Visual Paradigm versi 12.1. untuk spesifikasi alat bagi pengguna aplikasi adalah smartphone berbasis Android dengan minimal kebutuhan sistem resolusi 1920 x 1080 pada layar smartphone, versi minimal Android 5.1 'Lollipop', dan memiliki sensor Gyroscope.

Tahap ketiga adalah pengumpulan materi. Materi terapi Acrophobia yang berupa cara menyajikan simulasi didapat dari buku dan seorang psikiater. Musik latar belakang didapat dari internet yang disediakan oleh Youtube. Selain itu, pengumpulan gambar, model tiga dimensi dan elemen lainnya yang digunakan sebagai bahan pembuatan aplikasi dibuat dari perangkat lunak pendukung dan juga didapat dari internet. Penggunaan elemen banyak digunakan pada perancangan storyboard. Materi-materi tiga dimensi dibuat dengan aplikasi Autodesk Maya. Maya adalah software untuk keperluan 3D modeling, animasi, efek, serta solusi render yang telah digunakan dalam segala hal, mulai dari desain profuk sampai perfilman[7]. Antarmuka pengguna pada Maya terlihat bertambah dan berkembang sejak Autodesk terakhir dikembangkan. Antarmuka penggunaannya sangat sederhana dan dapat diubah. Labelnya jelas, petunjuk sangat lengkap, dan pembuatnya membuat Maya menjadi alat yang ramah[8]. Maya juga memiliki beberapa fitur lain untuk mengatur tampilan antarmuka terhadap pengguna[9].

Setelah materi terkumpul, lalu memasuki tahapan selanjutnya yaitu tahapan pembuatan. Pembuatan aplikasi ini menggunakan aplikasi Unity. Tahap keempat adalah pembuatan. Elemen-elemen yang sudah ada akan dikumpulkan dan dirangkai sesuai desain perancangan sistem yang sudah dibuat. Unity adalah pilihan sempurna untuk studio kecil, dan siapa pun yang ingin membuat game sendiri. Basis pengguna yang besar dan komunitas pengguna aktif yang memperbolehkan siapa pun, dari pemula sampai ahli, untuk mendapatkan jawaban dan membagikan informasi secepatnya[10]. Unity memiliki performa tinggi. Walaupun memiliki performa tinggi, tetapi *engine* ini stabil dan tidak terlalu berat, bahkan dapat tampil lebih baik saat runtime. Sehingga hanya dengan PC atau laptop dengan performa menengah sudah bisa membuat game[11]. Unity juga memiliki fitur membuat *terrain* yang lengkap[12].

Unity menggunakan bahasa pemrograman C# yang sederhana, modern, Object-Oriented Language, powerful dan fleksibel, efisien, modular, dan populer[13]. C# juga mendukung konstruksi kelas, antarmuka, struktur, dan enumerasi seperti bahasa

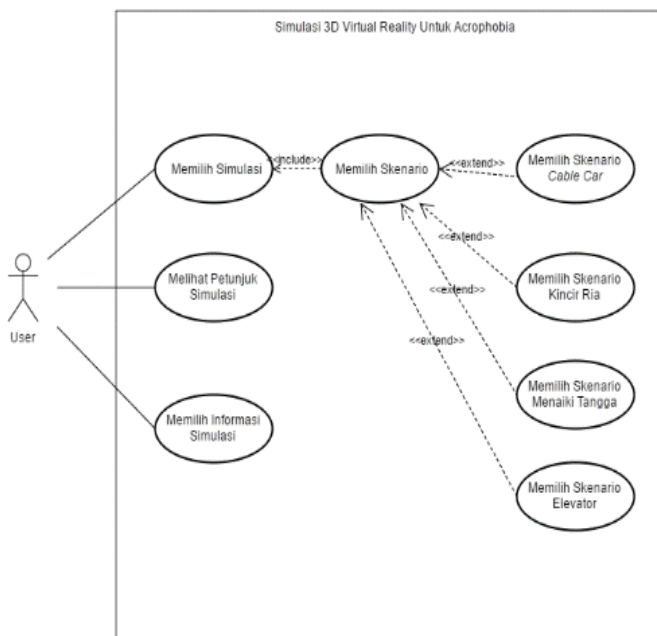
pemrograman lainnya[14] dan juga dapat terintegrasi dalam kelompok bahasa .NET lainnya[15].

Tahap kelima adalah pengujian. Pengujian aplikasi ini menggunakan metode kotak hitam (*black box*). Pengujian ini dilakukan untuk menunjukkan fungsi program yang dibuat tentang cara operasi dan kegunaannya, apakah keluaran data sesuai dengan yang diharapkan.

Tahap keenam adalah distribusi. Setelah melewati tahap pembuatan dan pengujian, maka aplikasi telah selesai dibuat. Aplikasi yang telah selesai dibuat akan didistribusi kepada psikiater yang menangani terapi acrophobia untuk diteliti lebih lanjut dalam bidang medis. Efek samping pada bidang medis tidak dibicarakan pada artikel ini.

Aplikasi ini dirancang pada sistem operasi Android dengan minimal versi Android 4.4 'Kitkat' dan dijalankan pada versi Android 5.1 'Lollipop'. Untuk uji coba aplikasi akan dilakukan pada Xiaomi Redmi Note 3 dengan sistem operasi Android versi Lollipop. *Smartphone* akan dikaitkan pada *cardboard* supaya dapat merasakan aplikasi Virtual Reality secara tepat.

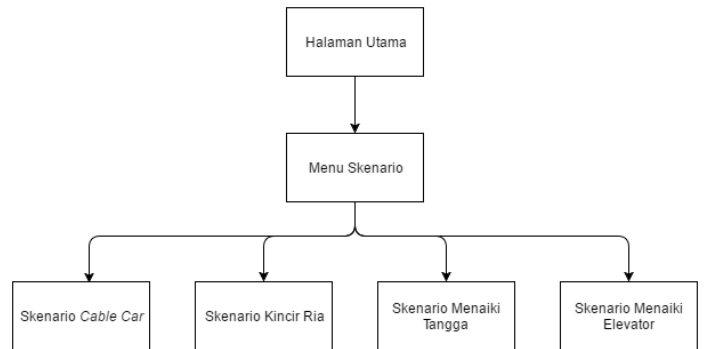
Perancangan sistem harus memperhatikan sasaran nantinya menjadi pengguna aplikasi ini, yaitu penderita *acrophobia* yang kebanyakan orang awam akan teknologi Virtual Reality. Oleh karena itu, perancangan sistem harus mudah digunakan oleh pengguna. Gambar 1 menunjukkan diagram *use case* untuk alur kerja aplikasi.



Gambar 1. Use Case Diagram Simulasi

Pembuatan alur simulasi yang menjelaskan alur terstruktur dari simulasi yang dirancang. Struktur navigasi simulasi dimulai ketika membuka aplikasi yaitu

tampilan halaman pembuka dan akan masuk ke halaman menu awal. Halaman menu awal memiliki satu menu pilihan dan dua panel informasi yaitu menu pemilihan skenario, panel instruksi dan panel tentang aplikasi. Struktur navigasi dari simulasi ini ditunjukkan pada Gambar 2.



Gambar 2. Struktur Navigasi Simulasi

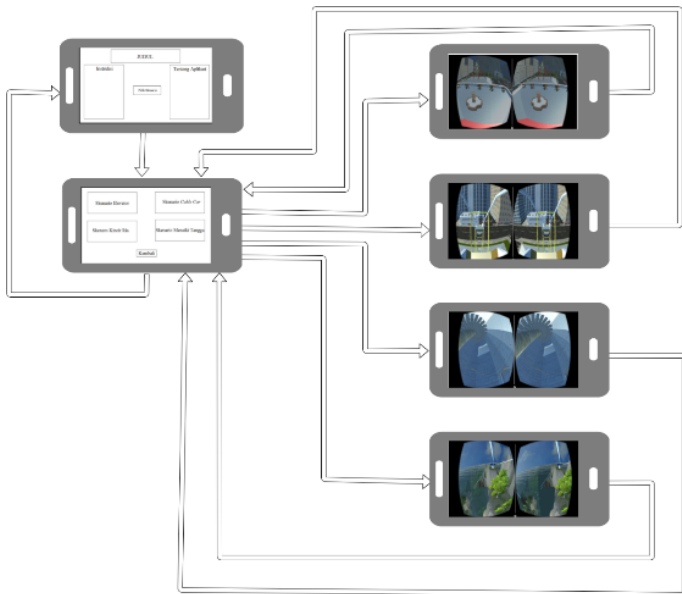
Langkah berikutnya adalah pembuatan perancangan tampilan aplikasi dengan menggunakan *storyboard* berupa tampilan *GameObject* yang digunakan untuk setiap halaman. Penjelasan penggunaan *storyboard* pada setiap halaman dilampirkan dalam Tabel 1.

Tabel 1. *Storyboard* halaman simulasi

<i>Storyboard</i> Halaman Utama	Merupakan halaman awal yang ditampilkan saat pertama kali aplikasi dijalankan.
<i>Storyboard</i> Halaman Pemilihan Skenario	Ditampilakn ketika tombol menu Play pada halaman utama ditekan. Halaman pemilihan skenario menampilkan skenario yang bisa dipilih oleh pengguna.
<i>Storyboard</i> Halaman Skenario Cable Car	Halaman ini merupakan tempat pengguna berada pada kegiatan menaiki <i>cable car</i> .
<i>Storyboard</i> Halaman Skenario Menaiki Kincir Ria	Halaman ini merupakan tempat pengguna berada pada kegiatan menaiki kincir ria.
<i>Storyboard</i> Halaman Skenario Menaiki Tangga	Halaman ini merupakan tempat pengguna berada pada kegiatan menaiki tangga.
<i>Storyboard</i> Halaman	Halaman ini merupakan tempat pengguna berada pada kegiatan

Skenario menaiki elevator.
 Naik Elevator

Masing-masing *storyboard* memiliki hubungan dengan *storyboard* yang lainnya. Berikut merupakan skema hubungan seluruh *storyboard* yang ada dalam permainan yang ditunjukkan pada Gambar 3.



Gambar 3. Skema Hubungan Antar *Storyboard*

III. HASIL DAN PEMBAHASAN

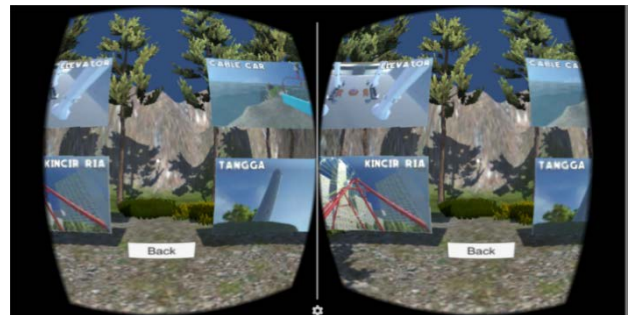
A. Hasil Tampilan

Halaman utama akan muncul pada saat pertama kali aplikasi dijalankan. Halaman ini akan memperlihatkan tata cara pemakaian aplikasi, tentang aplikasi, judul aplikasi, tombol ke pemilihan skenario serta memiliki lingkungan pepohonan, rumput, dan juga efek angin yang berhembus. Tampilan halaman utama aplikasi ditunjukkan oleh Gambar 4.



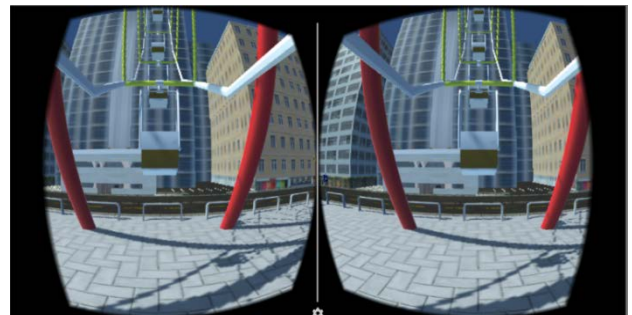
Gambar 4. Tampilan Halaman Utama

Halaman Pemilihan Skenario ditampilkan ketika tombol Play pada halaman utama ditekan. Tampilan halaman Pemilihan Skenario ditunjukkan oleh Gambar 5.



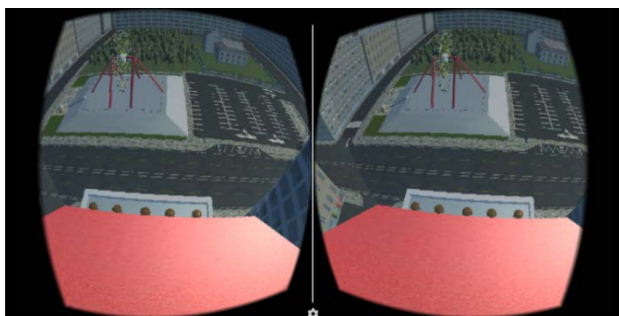
Gambar 5. Tampilan Halaman Pemilihan Skenario

Halaman Skenario Kincir Ria ditampilkan ketika tombol skenario kincir ria dipilih. Tampilan halaman Skenario Kincir Ria ditunjukkan oleh Gambar 6.



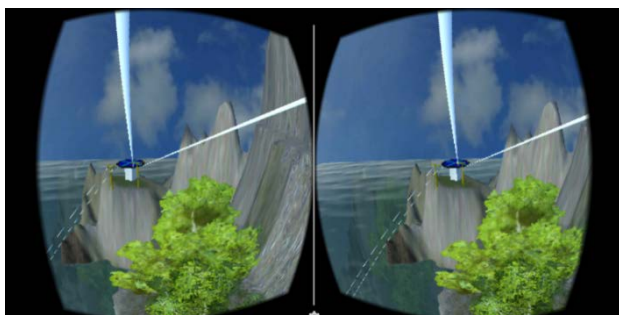
Gambar 6. Tampilan Halaman Skenario Kincir Ria

Halaman Skenario Elevator ditampilkan ketika tombol skenario elevator dipilih. Tampilan halaman Elevator ditunjukkan oleh Gambar 7.



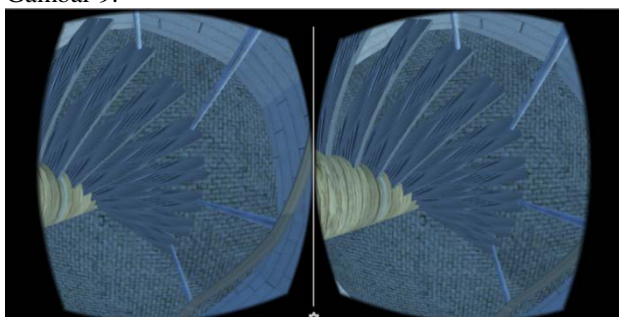
Gambar 7. Tampilan Halaman Skenario Elevator

Halaman Skenario *Cable car* ditampilkan ketika tombol skenario *cable car* dipilih. Tampilan halaman Skenario *Cable Car* ditunjukkan oleh Gambar 8.



Gambar 8. Tampilan Halaman Skenario *Cable Car*

Halaman Skenario Menaiki Tangga ditampilkan ketika tombol skenario tangga dipilih. Tampilan halaman Skenario Menaiki Tangga ditunjukkan oleh Gambar 9.

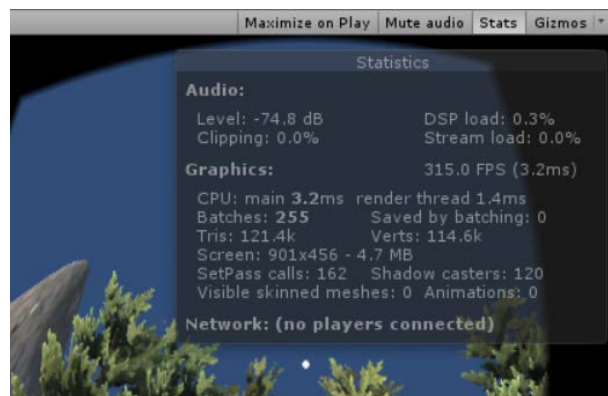


Gambar 9. Tampilan Halaman Skenario Tangga

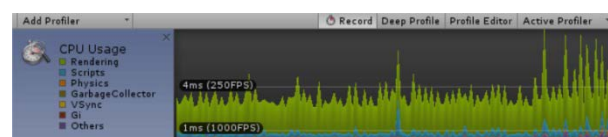
B. Pembahasan Hasil Penelitian

Pembahasan hasil penelitian meliputi pengecekan dan pengawasan performa aplikasi supaya dapat berjalan dengan optimal pada *smartphone*. Pengecekan performa

dapat dilihat di jendela Statistics pada Gambar 10 dan jendela Profiler seperti pada Gambar 11.



Gambar 10. Tampilan Stats pada Jendela Game

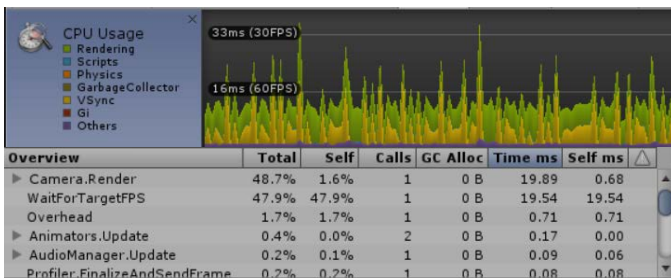


Gambar 11. Tampilan Jendela Profiler

Teknik pertama yang digunakan yaitu *static batching*. *Static batching* yaitu mengelompokkan objek-objek yang bertautan menjadi satu kesatuan yang besar. Teknik ini memungkinkan mesin untuk mengurangi banyaknya objek yang digambar pada layar dengan menyediakan material yang sama dalam suatu bidang dan tidak bergerak. Teknik ini lebih efisien dibandingkan dengan *dynamic batching* dalam mengendalikan performa, tetapi teknik ini memerlukan lebih banyak memori. *Static batching* membutuhkan memori tambahan untuk meletakkan bidang yang telah digabungkan. Jika objek-objek yang ada membagi bidang yang sama sebelum *static batching*, maka bidang yang lain akan dibuat tiap objek. Gambar 12 sampai Gambar 15 menunjukkan perubahan statistik elemen dan performa kecepatan *rendering* dari 19.89 ms menjadi 14.72 ms.



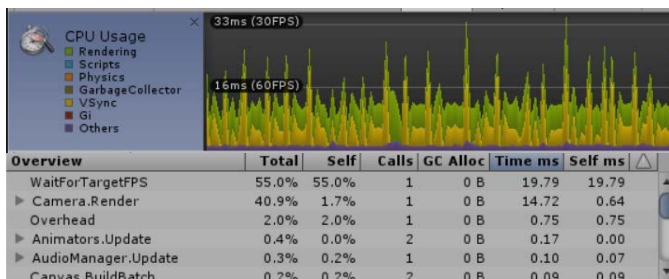
Gambar 12. Tampilan Stats Sebelum Melakukan *Static Batching*



Gambar 13. Tampilan Profiler Sebelum Melakukan *Static Batching*



Gambar 14. Tampilan Stats Setelah Melakukan *Static Batching*



Gambar 15. Tampilan Profiler Setelah Melakukan *Static Batching*

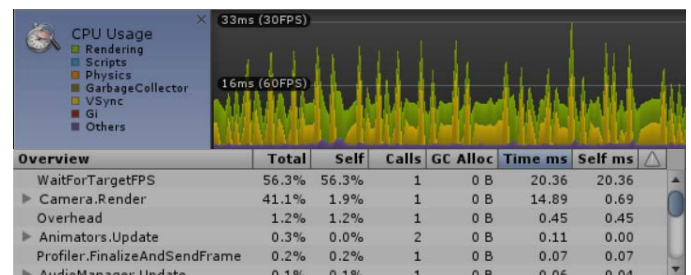
Teknik kedua yaitu *occlusion culling*. *Occlusion culling* adalah fitur yang menonaktifkan proses rendering pada objek yang tidak dilihat oleh kamera karena objek tersebut tertutup objek lain. Fitur ini tidak terjadi secara otomatis dalam pemrosesan 3D karena objek yang jauh dari kamera sering digambar pertama kali dan objek yang lebih dekat digambar setelahnya (biasa disebut dengan *overdraw*).

Proses *occlusion culling* akan melihat skenario menggunakan kamera virtual untuk membangun hierarki potensial objek yang terlihat. Data ini digunakan pada saat waktu runtime kamera untuk mengidentifikasi apa yang terlihat dan apa yang tidak terlihat. Berbekal informasi tersebut, Unity akan memastikan hanya objek yang terlihat yang akan dilakukan proses rendering. Hal ini membuat pemanggilan gambar berkurang dan meningkatkan performa.

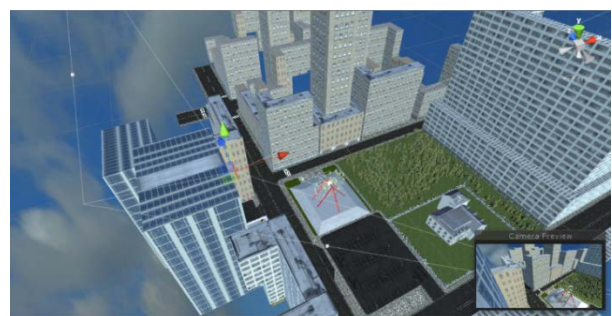
Data dari *occlusion culling* terdiri dari sel-sel yang tiap selnya adalah subdivisi dari seluruh ikatan volume dari skenario. Lebih khususnya sel-sel tersebut membentuk sebuah pohon biner. *Occlusion culling* mempunyai dua pohon, satu untuk objek yang statis dan yang lain untuk objek yang bergerak. Objek statis memetakan sel ke dalam daftar indeks yang didefinisikan sebagai objek yang terlihat, dimana yang mendapatkan hasil yang lebih akurat. Pada Gambar 16, Gambar 17 dan Gambar 18 menunjukkan statistik elemen, kecepatan *rendering* pada angka 14.89 ms, dan keadaan lingkungan sebelum melakukan teknik *occlusion culling*. Pada Gambar 19 menunjukkan keadaan lingkungan ketika proses *occlusion culling*. Pada Gambar 20, Gambar 21 dan Gambar 22 menunjukkan statistik elemen, kecepatan *rendering* pada angka 13.07 ms, dan keadaan lingkungan sesudah melakukan teknik *occlusion culling*.



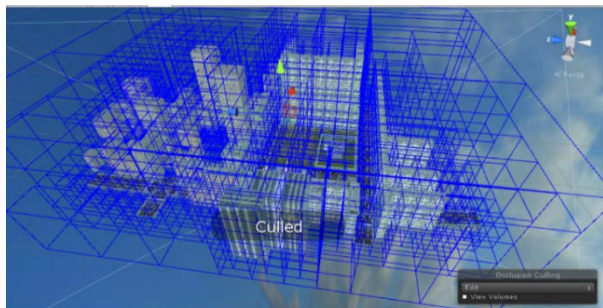
Gambar 16. Tampilan Stats Sebelum Melakukan *Occlusion Culling*



Gambar 17. Tampilan Profiler Sebelum Melakukan *Occlusion Culling*



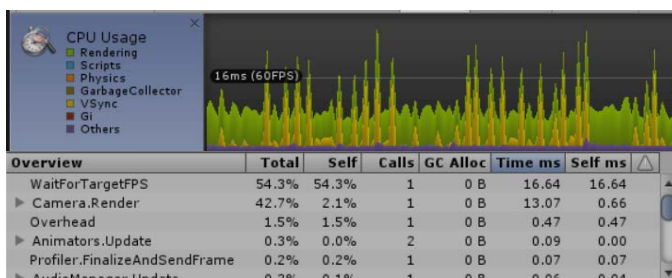
Gambar 18. Tampilan lingkungan sebelum melakukan *Occlusion Culling*



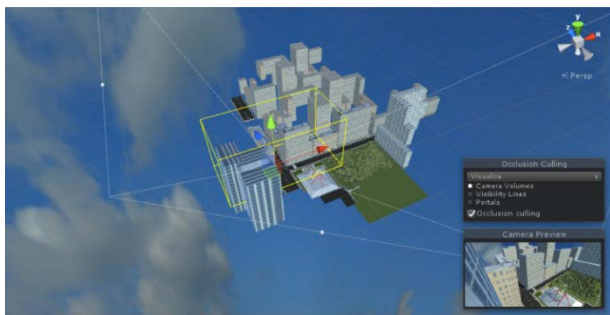
Gambar 19. Pemrosesan *Occlusion Culling*



Gambar 20. Tampilan Stats setelah melakukan *occlusion culling*



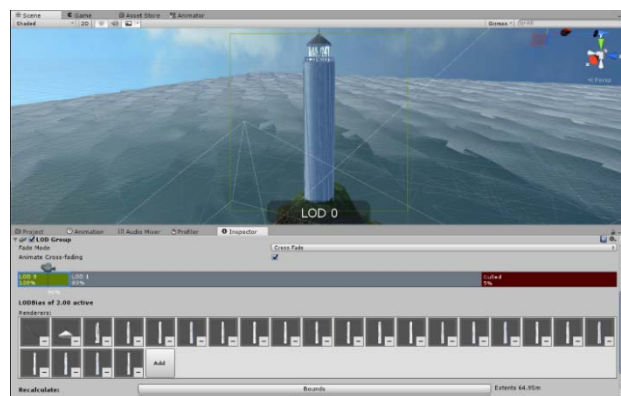
Gambar 21. Tampilan Profiler setelah melakukan *occlusion culling*



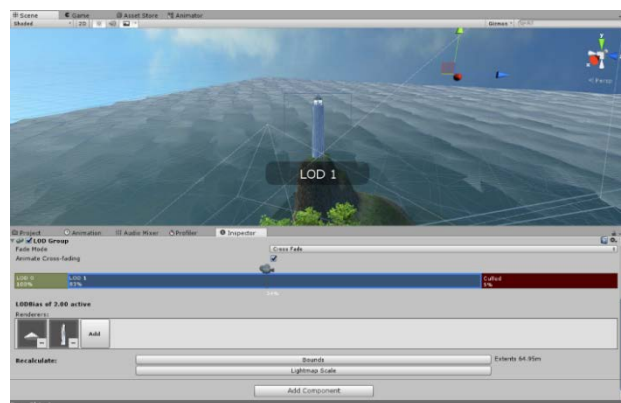
Gambar 22. Tampilan Lingkungan Setelah Melakukan *Occlusion Culling*

Teknik ketiga yaitu *level of detail* atau LOD. LOD adalah salah satu cara untuk mengatur kualitas

objek tergantung seberapa jauh kamera dengan objek tersebut dengan cara mengubah-ubah objek dengan objek dengan kualitas detail yang bagus sampai dengan kualitas detail yang buruk, serta dapat juga mengurangi komponen-komponen yang ada pada objek tersebut. Teknik ini menggunakan beberapa model ataupun satu model dengan detail yang berbeda untuk satu objek. Detail yang bagus dipakai jika kamera dekat dengan objek, sedangkan detail dengan kualitas yang kurang dipakai ketika kamera menjauh dari objek. Pengurangan kualitas tersebut bisa diganti dengan model lain dengan jumlah poligon yang lebih sedikit, atau pun mengurangi komponen pada model.



Gambar 23. Model yang Ditampilkan pada Sebuah Objek pada LOD 0 (Jarak Kamera Dekat)



Gambar 24. Model yang Ditampilkan pada Sebuah Objek pada LOD 1 (Jarak Kamera Jauh)

V. KESIMPULAN

Berdasarkan hasil pengujian pada aplikasi simulasi tiga dimensi Virtual Reality dengan mengoptimalkan teknik *rendering* elemen-elemen pada aplikasi, maka performa simulasi dapat berjalan dengan baik pada *smartphone* yang memiliki resolusi 1920 x 1080. Aplikasi yang terdiri dari empat skenario simulasi tersebut dapat di ekspor dan di instalasi dengan baik. Meskipun pemakaian *smartphone* dengan *cardboard* masih membuat mata pengguna kelelahan, tetapi semua

elemen yang ada pada aplikasi simulasi sudah dapat berjalan.

Berdasarkan kesimpulan diatas, dapat diketahui bahwa aplikasi ini masih dapat dikembangkan lebih jauh. Peneliti selanjutnya dapat menambah fitur skenario untuk mendiagnosis seberapa parah fobia yang pengguna derita. Selain itu dapat juga ditambahkan simulasi yang interaktif supaya pengguna tidak jenuh dan juga diperlukan penelitian lebih lanjut untuk mengetahui dampak klinis dari aplikasi terhadap pengguna. Pengembang juga dapat menganalisis efek dari penggunaan aplikasi terhadap penderita secara langsung dan meneliti lebih dalam pada bidang medisnya.

UCAPAN TERIMA KASIH

Terima kasih diucapkan kepada Dr. Innawati Jusup, M.Kes, SP.KJ yang telah membantu mengawasi pengembangan aplikasi dan memberikan masukan terhadap konten yang diperlukan dalam penelitian ini.

DAFTAR PUSTAKA

- [1] P. Halgin, Richard., Krauss Whitbourne, Susan. Psikologi Abnormal: Prespektif Klinis Pada Gangguan Psikologi. Salemba Humanika, 2009.
- [2] Jonathan Linowes. Unity Virtual Reality Projects. Packt Publishing Ltd, 2015.
- [3] Yao, Richard., Heath, Tom., Davies, Aaron., Forsyth, Tom., Mitchell, Nate., Hoberman, Perry. Oculus VR Best Practices Guide. Oculus VR, LLC, 2014.
- [4] Weghorst, Suzanne., Seibel, Eric., Oppenheimer, Peter., Hoffman, Hunter., Schowengerdt, Brian. Medical Interface Research at The HIT Lab. Springer-Verlag London Limited, 2008.
- [5] Repetto, Claudia., Gaggioli, Andrea., Pallavicini, Federica., Cipresso, Pietro., Raspelli, Simona., Riva, Giuseppe. Virtual Reality an Mobile Phones in The Treatment of Generalized Anxiety Disorders: a Phase-2 Clinical Trial. Springer-Verlag London Limited, 2011.
- [6] I. Binanto. Multimedia Digital-Dasar Teori dan Pengembangannya. Yogyakarta: C.V. ANDI OFFSET, 2010.
- [7] Goldstone, Will. Unity Game Development Essentials. Packt Publishing, 2009.
- [8] Ariatmanto, Dhani., Yoga Erlangga, Abisika. Penerapan visualisasi 3D dengan Atudeks Maya. Penerbit Andi, 2015.
- [9] Roy, Kenny., Rivera, Fiona. Sams Teach Yourself Maya in 24 Hours. Pearson Education, Inc, 2014.
- [10] Thorn, Alan. Mastering Unity Scripting. Packt Publishing Ltd, 2015.
- [11] Blackman, Sue. Beginning 3D Game Development with Unity. Lucasfilm Ltd, 2011.
- [12] Naas, Paul. Autodesk Maya 2014 Essentials. John Wiley & Sons, Inc, 2013.
- [13] Kurniawan, Agus. Pengenalan Bahasa C#. Project Otak, 2004.
- [14] Nugroho, Adi., Mengembangkan Aplikasi Basis Data Menggunakan C# dan SQL Server. Andi, Yogyakarta, 2010.
- [15] H, Erico Darmawan. Pemrograman Dasar Berorientasi Objek C#. Informatika, Bandung, 2014.